

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



# **Reconhecimento de Objetos Baseado em Visão Artificial**

**André Filipe Morais Duarte**

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Pedro Gomes da Costa

Co-orientador: José Magalhães Lima

28 de julho de 2015

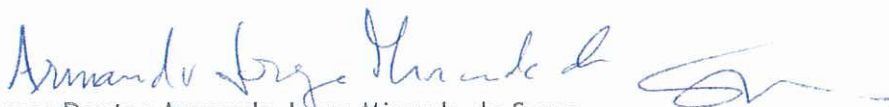


A Dissertação intitulada

“Reconhecimento de Objetos Baseado em Visão Artificial”

foi aprovada em provas realizadas em 16-07-2015

o júri



Presidente Professor Doutor Armando Jorge Miranda de Sousa  
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores  
da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Luís Paulo Gonçalves Reis  
Professor Associado do Departamento de Sistemas de Informação da Escola de  
Engenharia da Universidade do Minho



Professor Doutor Pedro Luís Cerqueira Gomes da Costa  
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores  
da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - André Filipe Morais Duarte



# Resumo

A competição de futebol robótico tem vindo a evoluir ao longo dos anos incentivando aos participantes a criação de sistemas cada vez mais complexos e capazes de resolver uma maior variedade de problemas.

Esta dissertação tem como objetivo a deteção e reconhecimento da bola de futebol por parte do guarda-redes tendo em conta que a bola de futebol pode ser de qualquer cor.

Os desafios que existem na deteção e reconhecimento de um objeto são muito variados como a falta de luminosidade, a própria velocidade de aquisição pois estando um objeto em movimento é necessário utilizar-se sistema com maior velocidade de aquisição de forma a permitir o acompanhamento do trajeto do objeto...

Para combater esses problemas inicialmente separou-se o problema da deteção da bola em dois casos sendo eles: a bola estar em contacto com o chão e o outro caso seria a bola não estar em contacto com o chão.

Para o caso da bola não estar em contacto com o chão separou-se a imagem de profundidade em *clusters* fornecendo assim uma segmentação da imagem baseada na distância, prosseguindo posteriormente para o reconhecimento dos objetos nos *clusters*.

Para o caso da bola em contacto com o chão realizou-se a segmentação do plano do chão o que permitiu isolar a bola do resto do ambiente seguida pela aplicação da técnica da segmentação por profundidade, avançando posteriormente para a fase de reconhecimento dos objetos nas diferentes imagens.

Na fase de reconhecimento considera-se que cada objeto das diferentes imagens são candidatos a ser uma bola de futebol. Recorrendo às características de uma bola de futebol o sistema é capaz de avaliar a presença no campo de visão do robô da bola de futebol.

Verificou-se que com o movimento rápido da bola que nem sempre é possível avaliar a presença da bola de futebol pois esta fica bastante deformada em algumas aquisições das imagens. Assim para combater estes erros de deteção desenvolveu-se um filtro de Kalman que oferece uma melhor estimativa da continuidade do movimento do objeto que se está a identificar ao longo do tempo.



# Abstract

The robotic football competition has evolved over the years to encourage participants to develop systems increasingly complex and able to address a wider variety of problems.

This thesis aims at detection and recognition of the soccer ball from the goalkeeper taking into account that the ball can be any color.

The challenges that exist in the detection and recognition of an object are varied from the lack of lighting, the speed acquisition, for instance if the object is in fast motion is needed to use a system with greater acquisition rate in order to track the path of the object, ...

It was initially separated the problem of the ball detection in two cases being: the ball is in contact with the ground floor and the other case the ball isn't in contact with the ground floor.

The case which the ball does not come in contact with the floor the depth image is separated into *clusters* thus providing a segmentation of the image based on the distance, subsequently proceeding to the recognition of objects in the *clusters*.

In the next case which the ball is in contact with the floor, the removal of the ground plane was essential to isolate the ball from the rest of the objects. The resulting image was followed by the application of the depth segmentation already explained, subsequently advancing to the recognition phase of the different *clusters*.

In the recognition phase it is considered that each one of the different objects are candidates for being a soccer ball. Using the characteristics of a soccer ball the system is able to evaluate the presence of such object in the robot line of sight.

It was found that the rapid movement of the ball that isn't always possible to evaluate the presence of the soccer ball because the motion of the ball can deformed the ball in some image acquisitions. Thus to combat these detection errors it was developed a Kalman filter that provides a best estimation of the object movement.





# Agradecimentos

Queria começar por agradecer à minha família por tudo o que me proporcionaram ao longo dos anos, deixar também uma palavra especial para os meus pais e irmã que sempre me apoiaram e me fizeram sempre seguir os melhores caminhos. O que sempre desejei foi deixar-vos orgulhosos e acho que vou conseguindo, pois vocês também são o meu orgulho.

Agradecer ao meu orientador, Pedro Costa, e coorientador, José Lima, por me terem aceitado para o desenvolvimento deste projeto e agradecer também todo o tempo dispensado para o acompanhamento que foi realizado ao longo do semestre de forma a incentivar-me a obter mais e melhores resultados. Uma palavra de agradecimento também pelo apoio e a disponibilidade do Doutor Andry Pinto que sempre se mostrou disponível para ajudar nos momentos de maior stress durante o desenvolvimento da dissertação.

Agradecer ao Doutor Cândido Duarte que me proporcionou uma das melhores experiências ao longo da minha formação, um estágio e a participação num projeto para a Infineon e a oportunidade de integrar um núcleo de estudantes que permitiu o meu melhor desenvolvimento como engenheiro e como pessoa.

Agradecer também ao Engenheiro Pedro Costa da Infineon por acreditar e apoiar sempre o meu trabalho e por me transmitir a sua experiência de engenheiro adquirida ao longo dos seus anos de trabalho.

Por fim, mas não os menos importantes, aos meus amigos de sempre e de faculdade deixo uma palavra de agradecimento por me terem acompanhado ao longo destes anos sem vocês não seria o mesmo.

André Morais



*“Faz por ser bom rapaz”*

José Morais



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento . . . . .	1
1.2	Motivação e Objetivos . . . . .	1
1.3	Estrutura da Dissertação . . . . .	2
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>3</b>
2.1	Introdução . . . . .	3
2.2	Sensores RGB-D . . . . .	3
2.2.1	Microsoft Kinect 1.0 . . . . .	4
2.3	Técnicas de Detecção e Reconhecimento de Objetos . . . . .	5
2.3.1	<i>Hough Voting</i> . . . . .	5
2.3.2	<i>Scale-Invariant Feature Transform</i> . . . . .	6
2.3.3	<i>Speeded Up Robust Feature</i> . . . . .	7
2.3.4	<i>Normal Aligned Radial Feature</i> . . . . .	8
2.3.5	<i>Geometric Hashing</i> . . . . .	10
2.3.6	<i>Random Sample Consensus</i> . . . . .	10
2.3.7	<i>Iterative Closest Point</i> . . . . .	11
2.3.8	<i>Point Feature Histograms</i> . . . . .	12
2.3.9	<i>Fast Point Feature Histograms</i> . . . . .	13
2.3.10	<i>Viewpoint Feature Histograms</i> . . . . .	14
2.3.11	<i>Feature Matching</i> . . . . .	15
<b>3</b>	<b>Algoritmo de Detecção</b>	<b>17</b>
3.1	Bolas Aéreas . . . . .	17
3.1.1	<i>Clustering</i> . . . . .	17
3.1.2	<i>K-Means Clustering</i> . . . . .	17
3.1.3	Resultados . . . . .	17
3.2	Bolas Não-Aéreas . . . . .	19
3.2.1	Segmentação do Campo baseado em cor . . . . .	19
3.2.1.1	RGB . . . . .	19
3.2.1.2	HSV . . . . .	20
3.2.1.3	YUV . . . . .	21
3.2.1.4	Resultados . . . . .	22
3.2.2	<i>Ground Plane</i> . . . . .	23
3.2.2.1	Transformações . . . . .	23
3.2.2.2	<i>RANSAC</i> . . . . .	24
3.2.2.3	Resultados . . . . .	24
3.2.3	<i>Clustering</i> . . . . .	25

3.2.4	Resultados . . . . .	25
3.3	Conclusões . . . . .	26
<b>4</b>	<b>Algoritmo de Reconhecimento</b>	<b>29</b>
4.1	Resultados . . . . .	32
4.2	Conclusões . . . . .	33
<b>5</b>	<b>Filtro de Kalman</b>	<b>35</b>
5.1	Modelo para Bola Aérea . . . . .	36
5.2	Modelo para Bola Não-Aérea . . . . .	37
5.3	Resultados . . . . .	38
5.4	Conclusões . . . . .	39
<b>6</b>	<b>Conclusões</b>	<b>41</b>
6.1	Conclusões Gerais . . . . .	41
6.2	Desenvolvimentos Futuros . . . . .	42
	<b>Bibliografia</b>	<b>43</b>

# Lista de Figuras

2.1	Microsoft Kinect 1.0. [1]	4
2.2	Diagrama de blocos do algoritmo proposto em [3].	5
2.3	Resultados da aplicação do algoritmo proposto por [3] onde a cena foi adquirida a partir de uma câmara <i>Kinect</i> e conseguiram a detecção de dois modelos na imagem (Coelho e o Robô).	6
2.4	Exemplo de Detecção utilizando <i>SIFT</i> . [4]	7
2.5	Os pontos de interesse detetados na imagem da esquerda (imagem de treino) são comparados com os pontos da imagem da direita (imagem adquirida pelo sistema). [6]	8
2.6	Identificação das bordas dos objetos. [7]	9
2.7	Identificação dos pontos de interesse para o descritor <i>NARF</i> . [7].	9
2.8	Exemplo do cálculo do descritor <i>NARF</i> (à direita) sobre um ponto de interesse (à esquerda). [7]	10
2.9	Comparação do método <i>RANSAC</i> com o método dos mínimos quadrados. [10]	11
2.10	Iterações do algoritmo ICP. [14]	12
2.11	Definição do referencial <i>uvw</i> em $p_s$ . Os ângulos $\alpha, \phi$ e $\theta$ descrevem as diferenças entre as normais $n_s$ e $n_t$ . [6]	13
2.12	Esquema da vizinhança do ponto $p_q$ . Onde o método FPFH é influenciado por 5 vizinhos. [6]	14
2.13	Componente de observação do algoritmo VFH. [6]	15
3.1	Imagem de profundidade com bola aérea.	18
3.2	<i>Clustering</i> da imagem de profundidade da figura 3.1.	18
3.3	Representação geométrica do espaço de cores RGB. [17]	20
3.4	Representação geométrica do espaço de cores HSV. [19]	21
3.5	Representação das componentes UV do espaço de cores YUV. [16]	22
3.6	Conversão e <i>threshold</i> da imagem RGB.	23
3.7	Esquema definido para obter as coordenadas reais a partir da posição dos diferentes pixels da imagem.	24
3.8	Segmentação do plano do chão.	25
3.9	Imagem de profundidade com bola não-aérea.	25
3.10	<i>Clustering</i> da imagem de profundidade da figura 3.9.	26
3.11	Algoritmo do Sistema.	27
4.1	Bola de futebol utilizada nos ensaios realizados.	29
4.2	Gráfico das variações das diferentes características com a distância à Kinect.	30
4.3	Algoritmo de Reconhecimento.	31
4.4	Resultado do Algoritmos abordado em 3.1 e 3.2.	32

5.1	O filtro de Kalman guarda os estados estimados do sistema e a incerteza dessa estimaco. A estimativa  atualizada recorrendo ao modelo transitrio e s medidas obtidas. $\hat{x}_{k k-1}$ corresponde  estimaco do estado do sistema no tempo k antes do das medidas em $y_k$ tenham importncia assim o valor de $P_{k k-1}$ corresponde  incerteza da estimaco. [23] . . . . .	36
5.2	Resultado do Filtro de Kalman na deteco da Bola Area e No Area. . . . .	38
5.3	Resultado do Filtro de Kalman na deteco da Bola Area. . . . .	39



# Lista de Tabelas

2.1	Kinect 1.0 Especificações. <a href="#">[2]</a> . . . . .	4
4.1	Funções que modelam as características da bola. . . . .	30
5.1	Erro médio da posição real da bola em comparação com a posição retornada pelo filtro de Kalman. . . . .	39



# Abreviaturas e Símbolos

3D	Tridimensional
DoG	Difference of Gaussian
FLANN	Fast Library for Approximate Nearest Neighbors
FPFH	Fast Point Feature Histograms
ICP	Iterative Closest Point
NARF	Normal Aligned Radial Feature
PCL	Point Cloud Library
PFH	Point Feature Histograms
RANSAC	Random Sample Consensus
RGB	Red-Green-Blue
RGB-D	Red-Green-Blue-Depth
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Feature
VFH	Viewpoint Feature Histograms



# Capítulo 1

## Introdução

### 1.1 Enquadramento

A evolução da tecnologia dos últimos anos tem permitido o aparecimento de sensores que possibilitam novos e melhorados métodos de interpretação do ambiente. Este facto é relevante para diversas áreas principalmente para a robótica, pois permitiu a criação e desenvolvimento de algoritmos que permitem perceber o mundo que nos rodeia.

No entanto estes algoritmos são normalmente computacionalmente pesados tornando-os inviáveis para sistemas robóticos baseados em visão artificial que requerem a extração de informação do mundo que os rodeia em tempo-real.

Desta forma o desenvolvimento e criação de algoritmos cada vez mais rápidos e eficientes têm sido um desafio para a comunidade científica.

### 1.2 Motivação e Objetivos

O objetivo desta dissertação passa pelo desenvolvimento de um sistema de deteção e reconhecimento de objetos com recurso a sensores RGB-D, mais propriamente, realizar o reconhecimento da bola de futebol num jogo de futebol robótico, que seja computacionalmente eficiente e ao mesmo tempo robusto a interferências do exterior assim como a luminosidade.

A motivação deste projeto é principalmente integrar todo o conhecimento adquirido na equipa de futebol robótico da FEUP (RoboSoccer-5DPO) e conseguir uma boa prestação numa futura competição de futebol robótico. Pois cada vez mais as competições estão mais competitivas e desafiantes o que incentiva estudantes a procurar novas alternativas que apresentem melhores resultados de forma a melhorar os seus sistemas.

### 1.3 Estrutura da Dissertação

A dissertação está dividida em seis capítulos, no capítulo 2 é apresentado a revisão bibliográfica, que consiste na análise de algoritmos já abordados que permitiram a resolução de problemas similares à dissertação.

No capítulo 3 é abordado o sistema de detecção que tem como objetivo a segmentação da imagem e identificar os diferentes objetos da imagem para serem posteriormente reconhecidos.

O sistema de reconhecimento da bola de futebol é especificado no capítulo 4 onde é especificado o algoritmo desenvolvido de forma a identificar os diferentes objetos como sendo uma bola de futebol.

No capítulo 5 é descrito a implementação e a incorporação do filtro de Kalman no sistema desenvolvido.

Por último apresenta-se as conclusões que foram possíveis retirar aquando a finalização do trabalho realizado.

## Capítulo 2

# Revisão Bibliográfica

### 2.1 Introdução

Neste capítulo aborda-se o estado da arte que identifica métodos e problemas que advém da tentativa do reconhecimento e identificação de objetos recorrendo a sensores RGB-D.

### 2.2 Sensores RGB-D

Os sensores RGB-D desde que foram introduzidos ao público com sucesso em meados de 2010 por parte da Microsoft Kinect despertou um elevado interesse por parte de várias áreas de estudo na utilização destes sensores.

Os motivos são variados como por exemplo são sistemas económicos em comparação com outras tecnologias como por exemplo câmaras baseadas na tecnologia *Time-of-Flight*.

Um sensor RGB-D é um sistema que incorpora vários componentes essenciais como:

- Câmara RGB que permite captar o mundo à sua volta como uma câmara fotográfica.
- Sensor de profundidade que permite ao sistema obter a distância que cada objeto está da câmara.

Como o interesse por esta tecnologia tem vindo a aumentar o mercado destes sensores acompanhou esta mesma tendência apresentando aos consumidores uma maior variedade de sensores com cada vez melhores câmaras e diferentes funcionalidades por preços acessíveis.

Apresenta-se na próxima secção as propriedades do sistema de aquisição utilizado no projeto de forma a identificar as características e limitações do aparelho utilizado.

### 2.2.1 Microsoft Kinect 1.0

A Microsoft Kinect foi a primeira a ter grande sucesso no lançamento desta tecnologia, com a Kinect 1.0 (Figura 2.1). Na tabela 2.1 apresenta-se as especificações da Kinect 1.0.



Figura 2.1: Microsoft Kinect 1.0. [1]

Consumo Energético	2.25W
Distância de Utilização	Desde 0.5m até 4.5m
Campo de Visualização	57° Horizontal
	43° Vertical
Sensor	RGB
	Depth
	2 * Microfone
Profundidade do tamanho da imagem	VGA (640x480) : 30 fps
Resolução espacial	3 mm
Resolução de profundidade	1 cm
Interface	USB
Ambiente de Operação	Interior
Dimensões	24.9 x 6.6 x 6.7 cm
Preço	147,09 (Amazon)

Tabela 2.1: Kinect 1.0 Especificações. [2]



## 2.3 Técnicas de Detecção e Reconhecimento de Objetos

De forma a conseguir detetar e reconhecer objetos numa imagem foram desenvolvidos ao longo dos anos vários algoritmos com o objetivo de resolver uma variedade de problemas.

Nas próximas secções aborda-se alguns desses algoritmos onde especifica-se o seu princípio matemático e também as vantagens e desvantagens da sua utilização.

### 2.3.1 *Hough Voting*

A transformada de Hough é uma das técnicas mais populares no universo da visão artificial, esta técnica foi inicialmente introduzida à comunidade científica como uma ferramenta capaz de detetar linhas numa imagem bidimensional e que posteriormente foi evoluindo permitindo a deteção de círculos e elipses numa imagem.

Este algoritmo a partir da utilização de pontos de interesse de uma imagem assim como cantos, bordas, etc. dos objetos gerar um esquema de votação que acumula os parâmetros dos objetos presentes na imagem e ao aplicar-se um determinado *threshold* aos picos deste acumulador consegue-se determinar a presença de um determinado objeto.

A extensão deste pensamento para um ambiente tridimensional é muito direto permitindo a deteção de planos numa dada nuvem de pontos. No entanto para o algoritmo ser capaz de lidar com as rotações e translações do objeto no espaço de estados de *Hough* resulta num elevado custo computacional (i.e.  $O(M*N^3)$ ), onde  $M$  é o número de pontos da nuvem de pontos e  $N$  é o número de intervalos de quantificação).

De forma a evitar este facto foi proposto por F. Tombari e L.Stefano em [3] uma nova abordagem ao algoritmo da transformada de *Hough* aplicada a um ambiente tridimensional. Um diagrama de blocos é apresentado na figura 2.2 onde explica o funcionamento do algoritmo proposto.

No início é necessário efetuar a aquisição da cena e a extração de pontos de interesse usando um detetor que se adequa à cena em análise. Posteriormente usa-se um descritor 3D de forma a aumentar a informação contida em cada ponto de interesse detetado, dada esta lista de pontos e a informação neles contida realiza-se um *matching* dos objetos que se pretende detetar com os presentes na imagem.

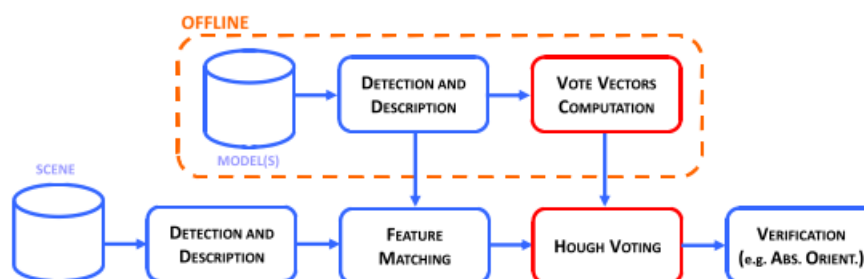


Figura 2.2: Diagrama de blocos do algoritmo proposto em [3].

Os resultados obtidos em [3] a partir deste algoritmo proposto são bastante satisfatórios como pode-se comprovar na análise da figura 2.3 onde a aquisição foi realizada por uma câmara Kinect

usando um descritor *SURF* 2.3.3 para a extração de pontos de interesse e um descritor *SHOT* para a projeção no espaço 3D do mapa de distâncias da cena. Para a determinação da pose do objeto foi utilizado o método *RANSAC* 2.3.6.

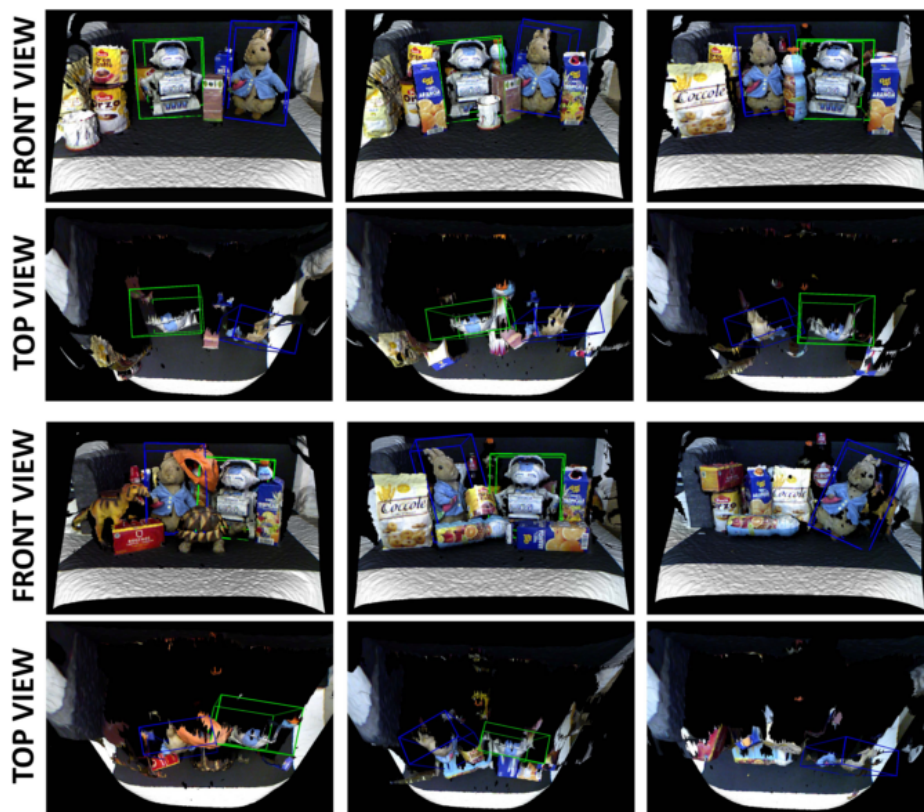


Figura 2.3: Resultados da aplicação do algoritmo proposto por [3] onde a cena foi adquirida a partir de uma câmara *Kinect* e conseguiram a deteção de dois modelos na imagem (Coelho e o Robô).

### 2.3.2 Scale-Invariant Feature Transform

O método *Scale-Invariant Feature Transform* mais conhecido por *SIFT* [4] é uma técnica de reconhecimento de objetos bastante utilizada em computação visual.

Para poder ser utilizado esta técnica é necessário existir um treino prévio onde se extraem as características dos objetos que não variam com a escala, rotações ou projeções no espaço 3D. Esta técnica tem a vantagem de conseguir combater os efeitos de diferenças na iluminação.

A eficácia deste algoritmo está dependente da qualidade das características que se extraem previamente e do facto de estas serem invariantes às transformações que a imagem esteja sujeita.

A extração das diferentes características é realizada iterativamente, onde primeiro aplica-se uma função gaussiana na direção horizontal a todas as linhas de pixéis e na seguinte operação na vertical em todas as colunas. Utiliza-se também uma pirâmide de imagens onde se realiza uma

progressiva interpolação bilinear com o objetivo de suavizar a imagem, sendo que a função gaussiana é aplicada a todas as camadas de modo a que cada uma seja comparada às suas adjacentes, de forma a conseguir determinar os máximos e mínimos. Na figura 2.4 apresenta-se um exemplo da detecção realizada pelo método *SIFT*. O resultado desta análise é um conjunto de vetores que representam as características do objeto.

$$g(x) = \frac{1}{\sqrt{2 * \pi * \sigma}} e^{-\frac{x^2}{2\sigma^2}} \quad (2.1)$$



Figura 2.4: Exemplo de Detecção utilizando *SIFT*. [4]

### 2.3.3 Speeded Up Robust Feature

O método *Speeded Up Robust Feature (SURF)* [5] é um algoritmo de reconhecimento de objetos que utiliza um princípio de extração prévia de pontos de interesse de um objeto recorrendo à análise da distribuição de intensidade de cores do objeto e estima características na região destes pontos utilizando um sistema de reconhecimento 2D.

Este reconhecimento de pontos de interesse é baseado no conceito de imagens integrais onde cada pixel  $x$  na imagem inicial é a soma dos valores dos pixels no retângulo formado pela origem da imagem e as coordenadas do pixel atual (2.2).

$$I_{\Sigma}(x) = \sum_{i \leq x} \sum_{j=0}^{j=0} I(i, j) \quad (2.2)$$

Este método é vantajoso ao *SIFT* pelo facto de ser mais robusto às transformações que a imagem pode sofrer e ao mesmo tempo consegue-se aumentar a *performance* da deteção nas imagens.

Na figura 2.5 demonstra-se a utilidade deste algoritmo no reconhecimento de pontos de interesse de uma garrafa de cerveja impressa numa caneca. Estes pontos de interesse serão posteriormente utilizados para identificar na base de dados os objetos que apresentam pontos de interesse similares.



Figura 2.5: Os pontos de interesse detetados na imagem da esquerda (imagem de treino) são comparados com os pontos da imagem da direita (imagem adquirida pelo sistema). [6]

O problema deste algoritmo é a utilização de um detetor de pontos de interesse 2D que não é capaz de distinguir se o que está a ser detetado é mesmo o objeto real ou está incorporado noutro objeto de diferentes dimensões, ou seja, neste caso na figura 2.5 é detetado a garrafa de cerveja mas não é esse objeto que está a ser observado mas sim uma caneca com a garrafa de cerveja impressa.

### 2.3.4 Normal Aligned Radial Feature

O algoritmo *Normal Aligned Radial Feature (NARF)* é uma técnica de reconhecimento de pontos de interesse a partir de uma imagem RGB onde a distância de cada pixel corresponde à intensidade da cor no espectro de luz visível, ou seja, objetos mais próximos são violeta e objetos mais distantes são vermelhos.

Após a aquisição da imagem de distância RGB é necessário identificar os pontos de interesse do descritor que se localizam nas bordas dos objetos. As bordas de um objeto podem ser divididas em três tipos:

- As bordas reais do objeto que estão localizadas nas pontas de cada objeto.
- As bordas provocadas pela sombra dos objetos.
- *Veil Points* são pontos interpolados das bordas anteriores que surgem num *scan* de um sensor LIDAR.

Ao eliminar-se as bordas que não interessam e identificando as reais bordas dos objetos consegue-se obter a figura 2.6.

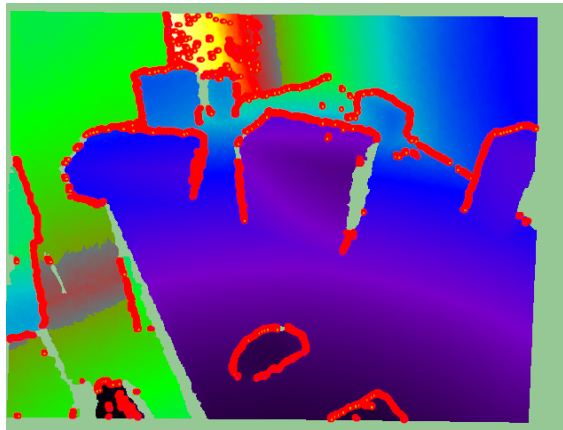


Figura 2.6: Identificação das bordas dos objetos. [7]

Posteriormente à extração das bordas dos objetos pode-se identificar os pontos de interesse da imagem, normalmente os pontos próximos dos cantos dos objetos são considerados bons pontos de interesse como pode-se verificar na figura 2.7.

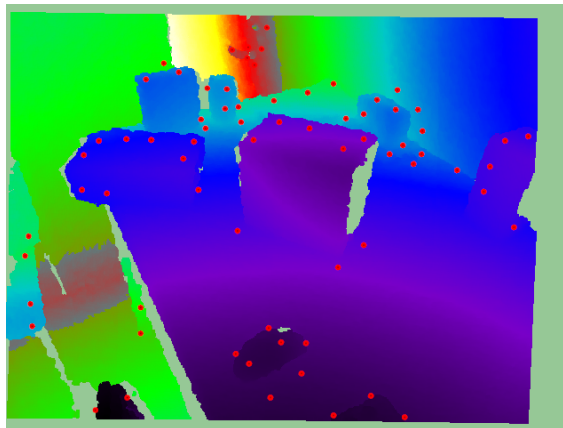


Figura 2.7: Identificação dos pontos de interesse para o descritor *NARF*. [7].

Para esta identificação o algoritmo trata de analisar cada borda atribuindo-lhe um determinado valor conforme a alteração da superfície ao seu redor. Posteriormente analisa-se a estabilidade do ponto, isto é, verifica-se se os pontos vizinhos a um determinado ponto são muito diferentes o que indica que a superfície altera-se muito nessa região logo o ponto é instável e um mau ponto de interesse.

Depois de aplicar todos os passos indicados é altura de se calcular o descritor *NARF*, este descritor transporta a informação sobre a superfície à volta de um dado ponto de interesse identificado anteriormente.

Primeiro é criado um *patch* local, uma parte da imagem que contém algum ponto de interesse como por exemplo bordas, cantos, etc., em torno de um ponto [8] seguidamente sobrepõem-se um padrão com forma de estrela com  $n$  raios sobre o *patch* centrado no ponto em análise como apresentado na figura 2.8. Calculando-se para cada raio um valor que reflete o quanto a superfície

se altera, ou seja, quanto maior a alteração da superfície e mais próximo do centro maior será o valor final.

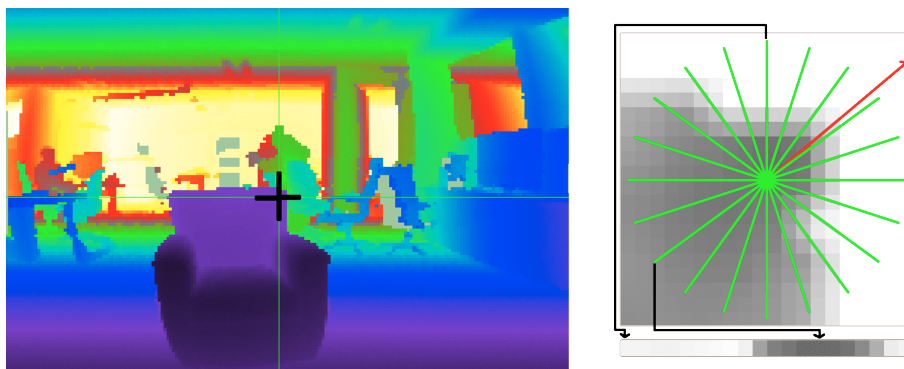


Figura 2.8: Exemplo do cálculo do descritor *NARF* (à direita) sobre um ponto de interesse (à esquerda). [7]

O resultado final do algoritmo é um descritor do objeto que é não invariante a rotações em torno da normal, que poderá ser comparado com os descritores dos objetos que se adquiriu anteriormente e que estão guardados em memória e proceder ao reconhecimento do objeto.

### 2.3.5 Geometric Hashing

O método *Geometric Hashing* é uma técnica baseada em modelos pré-existentes capaz de reconhecer objetos mesmo se estes são sujeitos a rotações, translações e/ou operações de escala.

Esta técnica através de pontos característicos dos objetos da cena comparar com a *hash table* construída previamente onde verifica se estes conjuntos de pontos se assemelham a algum modelo de objeto que se encontra na memória [9].

### 2.3.6 Random Sample Consensus

O algoritmo Random Sample Consensus (*RANSAC* [10]), é um algoritmo que através de um dado conjunto de dados extraí parâmetros do modelo matemático que definem as características dos objetos.

Estes parâmetros são obtidos iterativamente onde por cada iteração melhora-se os parâmetros obtidos até então. Esta técnica promoveu uma evolução ao método dos mínimos quadrados pois consegue evitar o efeito dos *Outliers* que muitas vezes danificam as estimativas dos parâmetros. Na figura 2.9 apresenta-se uma comparação do método dos mínimos quadrados com esta técnica.

Este método tem a desvantagem de poder ser lento variando conforme a presença de elevado número de *Outliers* e de parâmetros dos objetos. Este algoritmo é usado no reconhecimento de objetos utilizando a técnica de *feature matching* como proposto em [11].



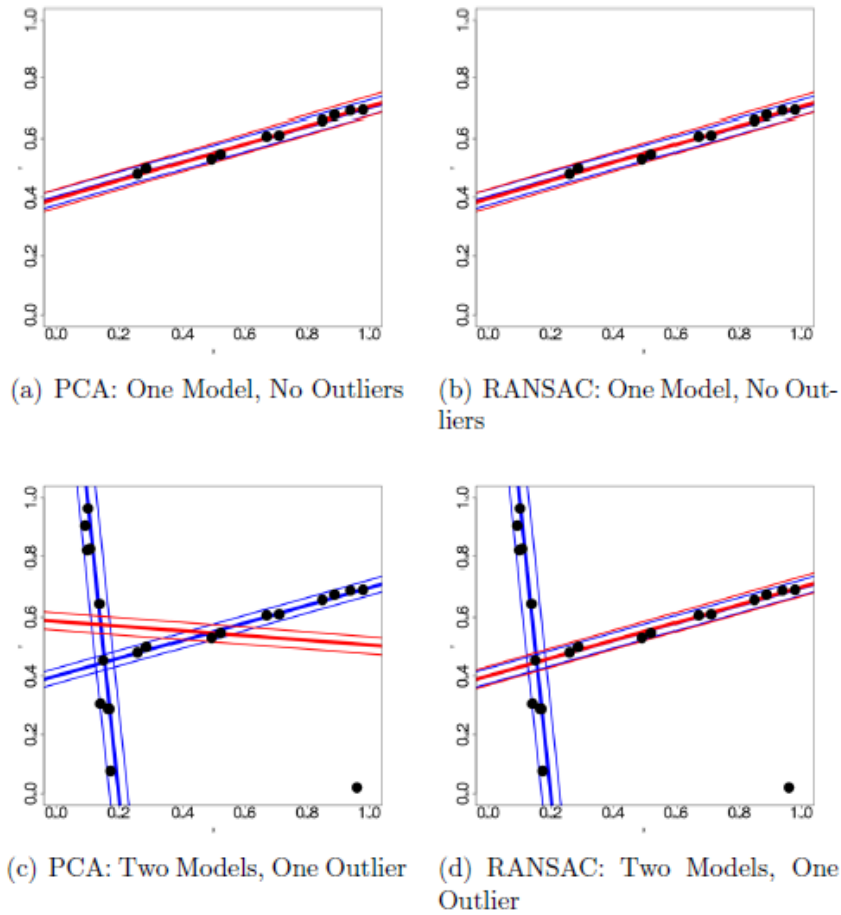


Figura 2.9: Comparação do método *RANSAC* com o método dos mínimos quadrados. [10]

### 2.3.7 Iterative Closest Point

A técnica *Iterative Closest Point* tem sido intensamente explorada desde o seu nascimento em 1992 [12] onde os investigadores tentam aumentar a rapidez do algoritmo, aumentar a robustez e descobrindo novas aplicações para este algoritmo.

O principal objetivo deste método é a partir de várias amostras obtidas do mesmo objeto/ambiente uni-las da melhor forma [13] ou seja sobrepô-las de forma a minimizar o erro 2.3 das translações e rotações entre as ligações. Na figura 2.10 demonstra-se um exemplo das iterações efetuadas pelo algoritmo ICP de modo a unir duas linhas.

$$E(R, t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \|m_i - (Rd_j + t)\|^2 \quad (2.3)$$

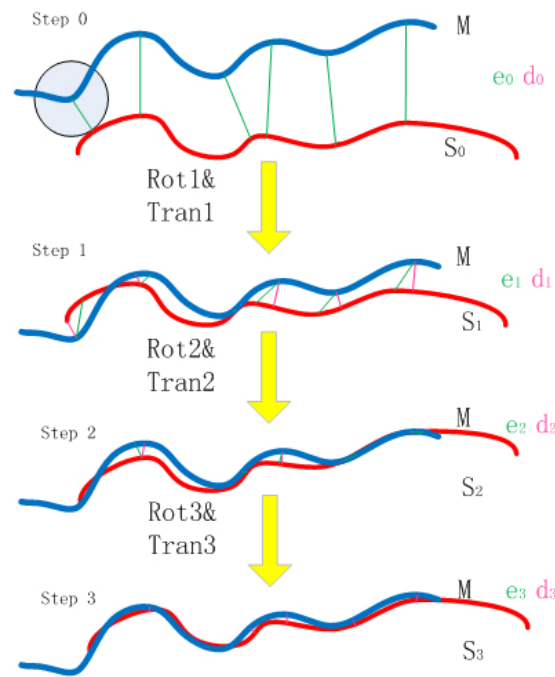


Figura 2.10: Iterações do algoritmo ICP. [14]

De forma a diversificar a utilização deste algoritmo existem vários métodos que permitem obter outros efeitos na utilização deste algoritmo como por exemplo: aumentar a velocidade do algoritmo recorrendo a *K-d trees*, evitar os mínimos locais removendo *Outliers*, usando processos estocásticos, entre outras hipóteses que vão variando conforme as necessidades.

Este algoritmo tem sido cada vez mais usado em aplicações como por exemplo Robôs de Busca e Salvamento, na inspeção de centrais de energia, na monitorização da linha costeira ou até numa área que está a ter maior relevância no mercado nos dias que correm que é a condução autónoma [12].

### 2.3.8 Point Feature Histograms

A técnica *Point Feature Histograms* (PFH) foi proposto por Rusu [6] em 2010. Esta técnica trata-se de um descritor de superfície para cada ponto de uma nuvem de pontos, a estimação de tal descritor requer o cálculo das normais as superfícies com o intuito de descrever as propriedades geométricas de cada ponto.

O primeiro passo da estimação é procurar os  $k$  vizinhos do ponto em questão ( $p$ ). Por cada par de pontos ( $p_s, p_t$ ) é calculado três ângulos  $\alpha, \phi$  e  $\theta$  que representam a relação entre cada ponto correspondendo as normais  $n_s$  e  $n_t$ . Assim define-se um referencial com origem em  $p_s$  utilizando



as expressões 2.4.

$$u = n_s \quad (2.4a)$$

$$v = u * \frac{p_t - p_s}{\|p_t - p_s\|_2} \quad (2.4b)$$

$$w = u * v \quad (2.4c)$$

Onde  $\|p_t - p_s\|^2$  representa a distância entre  $p_s$  e  $p_t$ .  $\alpha, \phi$  e  $\theta$  são definidos como mostra a figura 2.11 utilizando as equações 2.5.

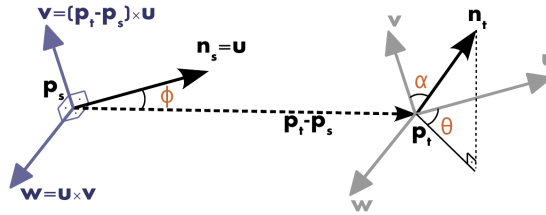


Figura 2.11: Definição do referencial  $uvw$  em  $p_s$ . Os ângulos  $\alpha, \phi$  e  $\theta$  descrevem as diferenças entre as normais  $n_s$  e  $n_t$ . [6]

$$\alpha = v * n_t \quad (2.5a)$$

$$\phi = u * \frac{p_t - p_s}{\|p_t - p_s\|_2} \quad (2.5b)$$

$$\theta = \arctan(w \cdot n_t, u \cdot n_t) \quad (2.5c)$$

No final divide-se as medidas de distância em  $q$  barras de igual tamanho criando um histograma tridimensional com o total de  $q^3$  pontos. No entanto a utilização desta técnica em todos os pontos de uma nuvem de pontos iria demorar cerca de  $O(n*k^2)$  [15] o que é insuportável para aplicações de tempo-real.

### 2.3.9 Fast Point Feature Histograms

O método *Fast Point Feature Histograms* (FPFH) é um descritor de superfície com base no descritor PFH (2.3.8) onde a maior diferença regista-se no tempo de execução que é reduzido de  $O(n*k^2)$  para  $O(n*k)$  [15] para uma nuvem de pontos com  $n$  pontos. Permitindo desta forma aplicações que requerem respostas em tempo-real e consegue obter resultados muito semelhantes ao descritor PFH.

Este algoritmo sofre uma alteração baseado no método *Simplified Point Feature Histogram* (SPFH) que em vez de analisar cada par de pontos na vizinhança do ponto  $p$  analisa só os pares de pontos que contêm o ponto  $p$ . Para balancear esta falta de relação entre cada ponto da vizinhança

do ponto é usado a equação 2.6.

$$FPFH(p_q) = SPFH(p_q) + \frac{1}{k} \sum_{i=1}^k \frac{1}{w_k} \cdot SPFH(p_k) \quad (2.6)$$

Onde  $w_k$  representa a distância entre o ponto em análise ( $p_q$ ) e o ponto vizinho ( $p_k$ ). Mostra-se também na figura 2.12 uma representação gráfica da implementação deste método.

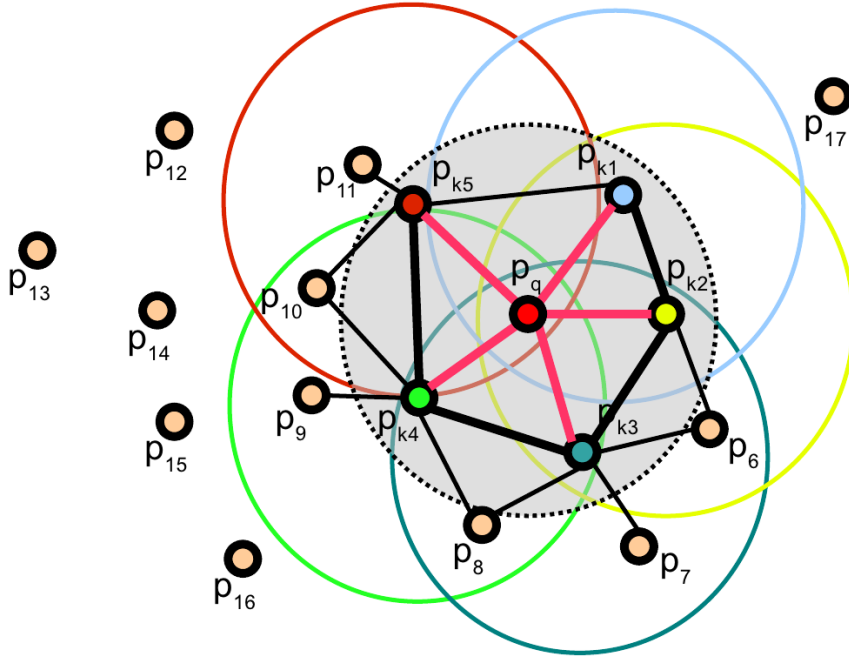


Figura 2.12: Esquema da vizinhança do ponto  $p_q$ . Onde o método FPFH é influenciado por 5 vizinhos. [6]

### 2.3.10 Viewpoint Feature Histograms

Esta técnica foi desenvolvida de forma a complementar o algoritmo FPFH (2.3.9) pois para além de identificar e reconhecer o objeto é necessário também a extração da sua pose, ou seja, a sua localização no espaço de acordo com um dado referencial. De modo a responder a este problema foi desenvolvido o método *Viewpoint Feature Histograms* (VFH) que agrupa o método FPFH (2.3.9) com um histograma de observação (na figura 2.13 apresenta-se um esquema da representação dos vetores que compõem o histograma).

Este histograma é composto pela distribuição dos ângulos entre todas as superfícies e o vetor localizado no centro de massa do objeto (de forma a evitar invariações de escala). Este método foi posto à prova por investigadores onde obtiveram uma exatidão de 98% [15].

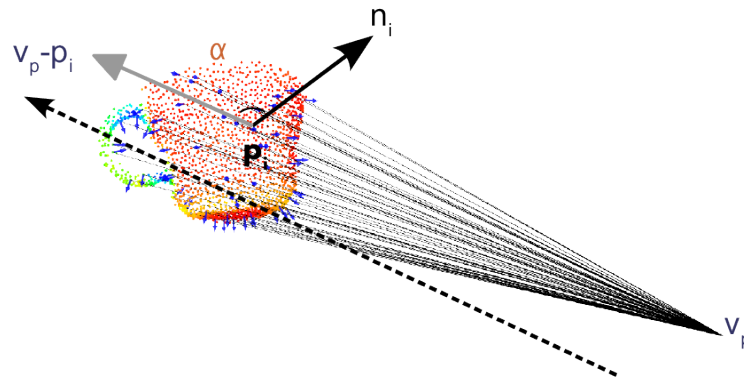


Figura 2.13: Componente de observação do algoritmo VFH. [6]

### 2.3.11 Feature Matching

O algoritmo de *Feature Matching* é responsável pela ligação entre o que se está a detetar e a informação que foi adquirida na fase de treino. A partir da decomposição dos histogramas em vetores de pontos de interesse pode-se aplicar a diferença entre eles recorrendo à norma Euclidiana (2.7) e verificar qual dos modelos de objetos têm menor erro de *matching* com o objeto que se observa da cena.

$$d(\vec{a}, \vec{b}) = \|\vec{a} - \vec{b}\| = \sqrt{\sum_{i=1}^m (a_i - b_i)^2} \quad (2.7)$$

No entanto esta norma dá mais valor aos pontos de valor aproximado a zero para contrapor este facto recorre-se à norma *Chi-square* pois normaliza os valores do histograma, onde pode ser calculado pela equação 2.8.

$$d_{Chi}(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^m \frac{(a_i - b_i)^2}{a_i + b_i}} \quad (2.8)$$

Foi comparado as diferente métricas dentro do contexto do FPFH (2.6) no entanto os resultados são pouco detalhados e contra intuitivos onde a norma de *Manhattan* (2.9) é a que apresenta melhor exatidão [6].

$$d_{Manhattan}(\vec{a}, \vec{b}) = \sum_{i=1}^m |a_i - b_i| \quad (2.9)$$



## Capítulo 3

# Algoritmo de Detecção

Neste capítulo apresenta-se as técnicas utilizadas para efetuar a deteção da bola de futebol. Observa-se que durante um jogo de futebol a bola pode estar presente no chão ou no ar e estes dois casos apresentam propriedades diferentes que podem ser vantajosas para o projeto.

Desse modo decidiu-se seguir essa filosofia e detetar a bola em dois diferentes casos, nas secções seguintes apresenta-se os processos utilizados para conseguir eficazmente realizar a deteção da bola de futebol.

### 3.1 Bolas Aéreas

#### 3.1.1 *Clustering*

De forma a separar os diferentes objetos na imagem de profundidade optou-se por separar a imagem em diferentes clusters aplicando a técnica de *K-Means Clustering* que é brevemente explicado de seguida.

#### 3.1.2 *K-Means Clustering*

O algoritmo *K-Means Clustering* é um algoritmo de quantificação de um vetor e tem como objetivo separar todas as observações e agrupa-las num número definido de cluster onde um ponto pertence a um dado grupo de pontos minimizando a soma dos quadrados (3.1) desse mesmo cluster.

$$\arg_s \min \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \quad (3.1)$$

#### 3.1.3 Resultados

Os resultados obtidos podem ser observados na figura 3.2, onde foi possível separar os diferentes elementos da figura original (3.1) de tal modo que na subfigura 3.2d observa-se que a bola fica isolada de todos os outros elementos permitindo seguir para a fase de identificação para de

facto comprovar que se trata do objeto que se quer identificar, este assunto de reconhecimento é tratado no capítulo 5.

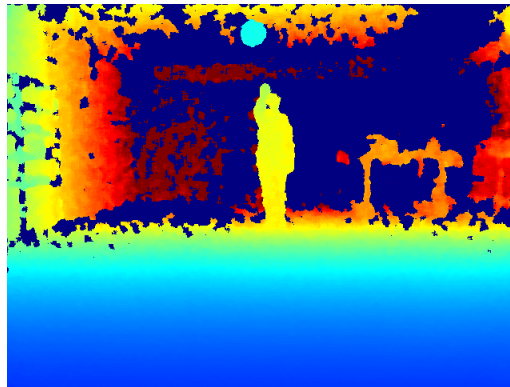


Figura 3.1: Imagem de profundidade com bola aérea.



(a) Cluster 1.



(b) Cluster 2.



(c) Cluster 3.



(d) Cluster 4.



(e) Cluster 5.

Figura 3.2: *Clustering* da imagem de profundidade da figura 3.1.

## 3.2 Bolas Não-Aéreas

Para a detecção de bolas que não são aéreas não é vantajoso usar a técnica usada nas bolas aéreas sem nenhum pré-processamento pois é necessário um maior número de *clusters* de forma a obter resultados similares às bolas aérea.

Este aumento do número de *clusters* irá prejudicar o tempo computacional que é necessário evitar pois irá reduzir o tempo de reação por parte do guarda redes, aumentando as hipóteses de sofrer golo.

Assim foi adotado uma técnica que nos forneceu resultados similares como nas bolas aéreas, o que nos permite usar o mesmo sistema de reconhecimento para verificar a existência da bola de futebol.

### 3.2.1 Segmentação do Campo baseado em cor

Recorreu-se à segmentação baseada em cor para identificar o campo de futebol pois este irá ser sempre verde independentemente dos robôs, bola ou outros elementos externos.

Assim é necessário converter a imagem que a Kinect nos fornece que utiliza o esquema de cores RGB dificultando a identificação dos pontos "*verdes*".

Este esquema de cor (RGB) é altamente afetado pela luminosidade da cena dificultando a escolha de valores para realizar um *threshold* que consiga eficientemente separar a cor que se pretende identificar.

Nas seguintes secções é explicado as vantagens e desvantagens dos esquemas de cores que merecem maior atenção para este projeto pois oferecem melhores hipóteses de sucesso.

#### 3.2.1.1 RGB

O espaço de cor RGB é utilizado pela grande maioria das câmaras digitais na aquisição de imagens do mundo que nos rodeia pois é também como os seres humanos veem o mundo, ou seja, numa combinação de três cores que habitualmente chamam-se primárias que são Vermelho (*Red-R*), Verde (*Green-G*) e Azul (*Blue-B*).

Posteriormente as imagens poderão ser transformadas em diferentes espaços de cores onde cada um tem as suas vantagens e desvantagens. A escolha do espaço de cor a utilizar é ainda um grande desafio para a segmentação baseada em cor.

O espaço de cores RGB pode ser geometricamente representado por um cubo tridimensional (figura 3.3). Onde as coordenadas de cada ponto dentro do cubo representa os valores do componente vermelho, verde e azul respetivamente [16].

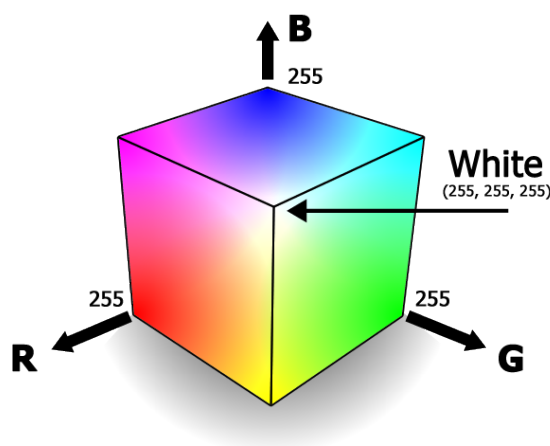


Figura 3.3: Representação geométrica do espaço de cores RGB. [17]

As regras de colometria são:

- Qualquer cor pode ser criada a partir de uma combinação única destas três componentes.
- Se duas cores são equivalentes estas manter-se-ão equivalentes após uma multiplicação ou divisão das três componentes pelo mesmo número.
- A iluminação da mistura das diferentes cores é igual à soma da luminância de cada cor separadamente.

Sendo assim este espaço de estados é o mais indicado para apresentação de cor pelas razões indicadas anteriormente no entanto não é uma representação útil para segmentação porque existe uma elevada correlação entre as diferentes componentes do espaço de cores, ou seja, se a intensidade alterar as três componentes vão sentir essas variações. [18]

### 3.2.1.2 HSV

O espaço de cor *HSV* (*Hue-Saturation-Value*) é uma das variações do sistema *HSI* (*hue-saturation-intensity*) existem outras variações como a representação *HSL* (*Hue-Saturation-Lightness*), *HSB* (*Hue-Saturation-Brightness*).

Este sistema separa a informação da cor de uma imagem da sua intensidade, onde a informação da cor é representada pelos valores de *Hue* e *Saturation* enquanto que a intensidade indica o brilho, ou seja, a quantidade de luz da cena de uma imagem é descrita na última componente [18].

A componente *Hue* indica a cor básica dos objetos baseando-se no comprimento de onda dominante da distribuição espectral de luz visível. A *Saturação* é a medida da pureza da cor, ou seja, significa a quantidade de luz branca misturada com a componente *Hue*.

O espaço de cor do sistema *HSV* pode ser descrito na figura 3.4. Desta forma a componente *Hue* é considerada como um ângulo entre a linha da referência e o ponto da cor no espaço RGB, variando assim entre os  $0^\circ$  e os  $360^\circ$ . A componente *Saturation* representa a distância radial desde o centro do cilindro, ou seja, quanto mais perto do centro a cor será mais clara. *Value* é a altura do cilindro que indica a intensidade dessa cor. Para a utilização deste espaço de cores é necessário efetuar os cálculos que estão indicados na equação 3.2 que permitem a mudança do espaço de



cores de RGB para HSV.

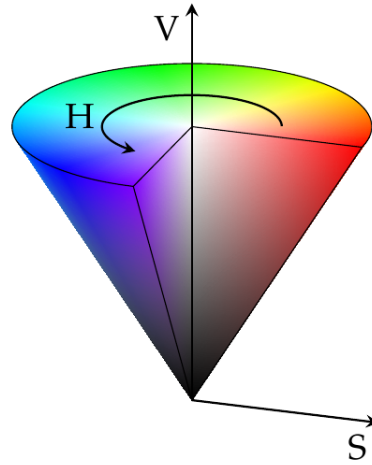


Figura 3.4: Representação geométrica do espaço de cores HSV. [19]

$$\left\{ \begin{array}{l} H = \begin{cases} \frac{60(G-B)}{S} & \text{if } V = R \\ 180 + \frac{60(B-R)}{S} & \text{if } V = G \\ 240 + \frac{60(R-G)}{S} & \text{if } V = B \end{cases} \\ S = \frac{255(V - \min(R, G, B))}{V} \\ V = \max(R, G, B) \\ \text{if } H < 0 \text{ then } H = H + 360 \end{array} \right. \quad (3.2)$$

Este espaço de cor apresenta várias propriedades que são boas para a realização de segmentação de objetos baseando-se em cor pois consegue-se de certa forma combater os efeitos da iluminação e sombras [18]. No entanto existem singularidades próximas do eixo do cilindro onde não se pode simplesmente analisar o canal da cor, *Hue*, para a distinção das várias cores o que conduz a que as restantes componentes do sistema também contribuam para a atribuição da cor.

### 3.2.1.3 YUV

O espaço de cor *YUV* foi desenvolvido de modo a melhorar o tratamento de imagens de vídeo aquando a utilização de algoritmos digitais. A componente *Y* transporta a característica *luma* que representa a luminosidade da imagem, e duas componentes de crominância (*UV*), sendo cada componente independente uma das outras [20].

A representação do espaço de cor YUV é diferente das restantes pois só as componentes UV transportam informação da cor e a componente Y informa quantidade de luminosidade da imagem, desta forma representa-se na figura 3 a representação das componentes UV. Para executar

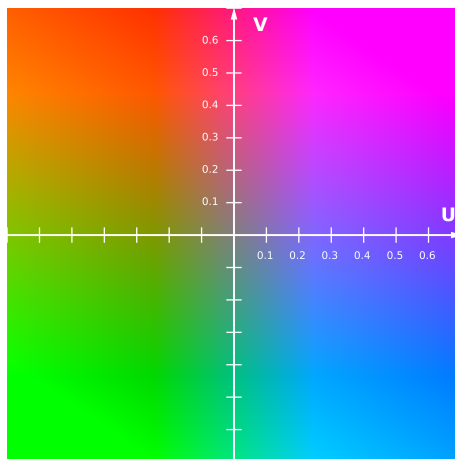


Figura 3.5: Representação das componentes UV do espaço de cores YUV. [16]

a conversão do espaço de cores RGB para o espaço de cores YUV é necessário efetuar os cálculos indicados em 3.3, sabendo que os valores das componentes RGB vão desde 0 até 1 e no resultado final Y está também no intervalo de 0 a 1, a componente U desde -0.436 até +0.436 e a componente V encaixa no intervalo de -0.615 até +0.615.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.3)$$

#### 3.2.1.4 Resultados

Para o presente projeto preferiu-se usar o espaço de cores YUV pois permite utilizar valores de *threshold* mais curtos em comparação do espaço de cores HSV e desta forma evitar a presença de outros elementos da imagem que possam interferir com o resultado que se quer atingir.

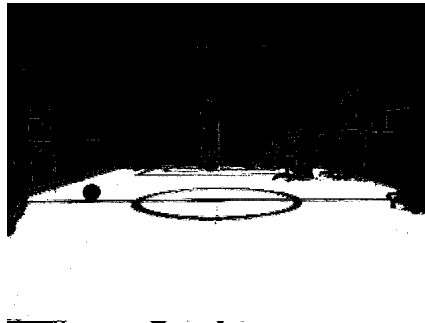
O resultado que se obteve pode ser observado na figura 3.6 onde na imagem da esquerda apresenta-se a imagem RGB (3.6a) e à direita a sua transformação no espaço YUV (3.6b). Por último apresenta-se na última figura (3.6c) a segmentação do relvado, ou seja, da cor verde.



(a) Imagem RGB.



(b) Imagem YUV.



(c) Imagem YUV segmentada.

Figura 3.6: Conversão e *threshold* da imagem RGB.

### 3.2.2 Ground Plane

Após obter-se os pontos que pertencem ao plano do chão pode-se calcular os parâmetros deste plano de modo a eliminar o chão e aplicar-se a mesma técnica que foi utilizada na detecção de bolas aéreas.

#### 3.2.2.1 Transformações

Para ser possível calcular os parâmetros do plano correspondente ao chão é necessário para os pontos que foram selecionados da imagem RGB obter a sua correspondente profundidade.

Para isso é necessário efetuar o mapeamento dos pixéis da imagem RGB para a imagem da profundidade usando os parâmetros da calibração realizar operações de translação e rotação de forma a obter os pixéis correspondentes na imagem de profundidade [21].

Posteriormente é necessário transformar estes pontos já na imagem da profundidade para os pontos nas coordenadas do mundo real para que o cálculo do plano não seja afetado pela deformação das lentes.

Deste modo é necessário efetuar as equações 3.4, onde  $(x_w, y_w, z_w)$  é as coordenadas no mundo real do ponto que no ecrã têm as coordenadas  $(x_v, y_v, f)$  e os valores de  $h$  e  $w$  são os tamanhos da

imagem (altura e largura respetivamente). Na figura 3.7 apresenta-se o esquema que foi possível extrair as equações 3.4 para determinar as coordenadas do ponto no mundo real [21].

$$\begin{cases} f = \frac{h}{2 \cdot \tan(\frac{43}{2})} & \text{ou} & f = \frac{w}{2 \cdot \tan(\frac{57}{2})} \\ x_w = z_w \cdot \frac{x_v}{f} \\ y_w = z_w \cdot \frac{y_v}{f} \end{cases} \quad (3.4)$$

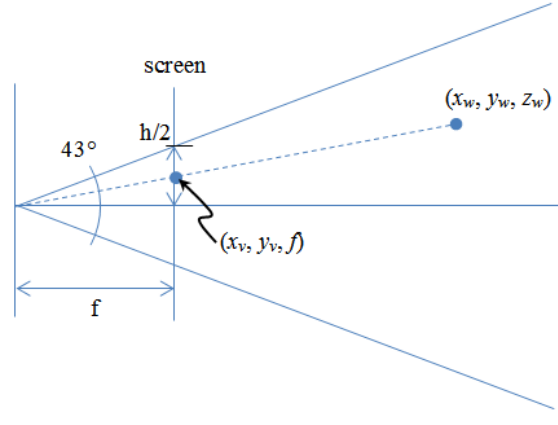


Figura 3.7: Esquema definido para obter as coordenadas reais a partir da posição dos diferentes pixéis da imagem.

### 3.2.2.2 RANSAC

Para efetuar o cálculo do plano do chão recorreu-se à técnica RANSAC já brevemente exposta em 2.3.6 que é uma técnica que aproxima uma série de pontos a um modelo matemático específico, que neste caso é utilizado o modelo matemático de um plano 3.5.

$$ax + by + cz + d = 0 \quad (3.5)$$

Após obter a equação do plano é obtida a componente da profundidade de cada ponto da imagem e é calculado a distância desses pontos ao plano calculado anteriormente. Através da equação 3.6 é possível calcular essa distância  $D$  ao plano para um ponto genérico  $(x_0, y_0, z_0)$ .

$$D = \frac{ax_0 + by_0 + cz_0 + d}{\sqrt{a^2 + b^2 + c^2}} \quad (3.6)$$

### 3.2.2.3 Resultados

Devido à presença de ruído na imagem da profundidade foi considerado que os pontos pertencem ao plano do chão caso estejam dentro de um intervalo definido. Este intervalo foi otimizado ao longo de uma série de testes resultando na segmentação apresentada na figura 3.8.

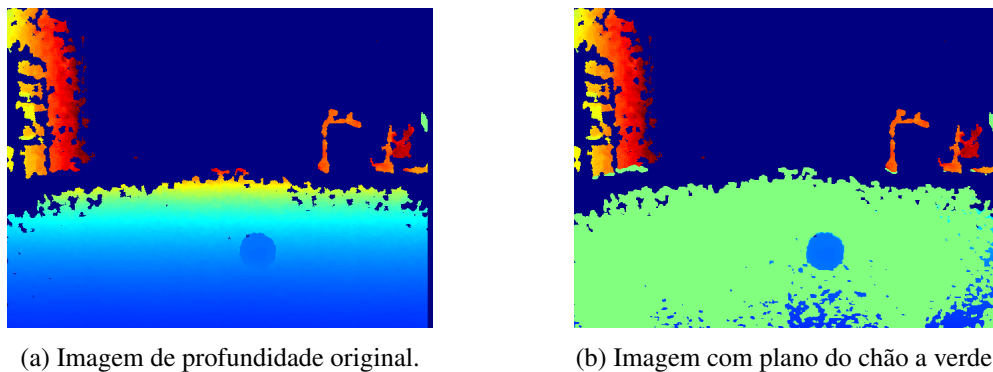


Figura 3.8: Segmentação do plano do chão.

### 3.2.3 Clustering

Após a remoção do plano do chão observou-se que existem pequenos pontos que podem conduzir à deformação dos objetos conduzindo à falha do algoritmo de reconhecimento que irá ser detalhado em 4.

De forma a reduzir esse fenómeno foram realizadas operações morfológicas [22], aberturas e fechados, à imagem de forma a remover certos erros de medição e ruídos indesejáveis e proceder à realização da técnica abordada em 3.1.2.

### 3.2.4 Resultados

Com esta abordagem obteve-se os resultados apresentados na figura 3.9 onde conseguiu-se separar as componentes que não interessam para a aplicação e realizar a segmentação da imagem de profundidade apresentada na figura 3.10.

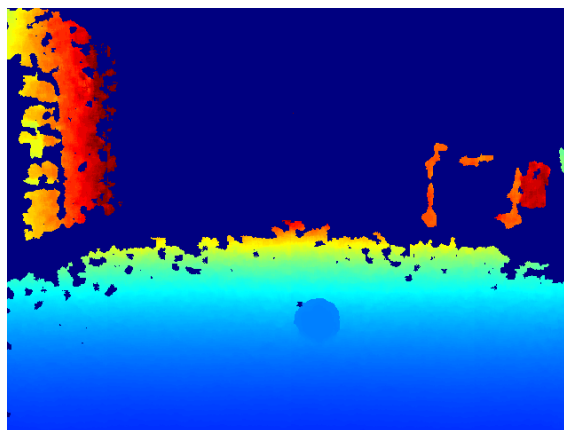


Figura 3.9: Imagem de profundidade com bola não-aérea.

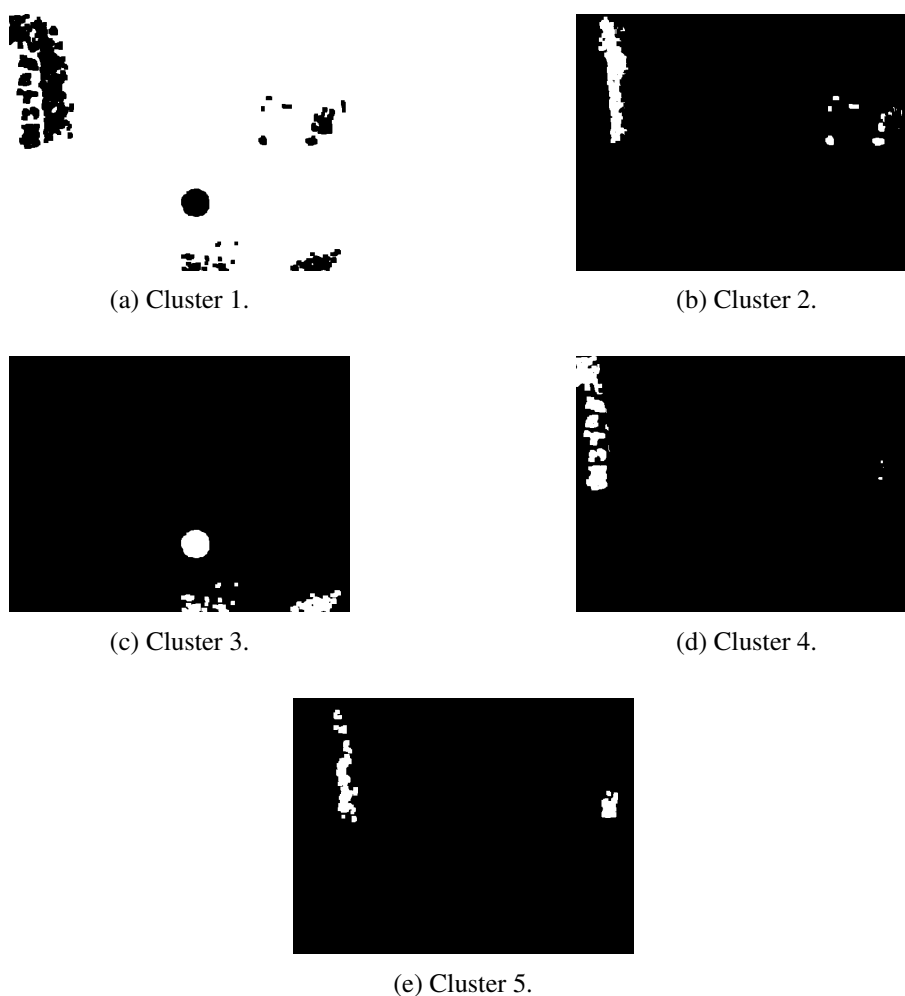


Figura 3.10: *Clustering* da imagem de profundidade da figura 3.9.

### 3.3 Conclusões

Em suma observou-se que os métodos utilizados neste capítulo são capazes de segmentar a imagem de profundidade permitindo observar a presença da bola num dos *clusters* obtidos.

Concluiu-se também que o número de *clusters* utilizados pode ser variado, o que no entanto afetará o tempo de processamento. A opção pelos cinco *clusters* foi tomada devido ao facto de ter um tempo de processamento reduzido ( $\sim 30\text{ms}$ ) e ao mesmo tempo permitir a uma boa segmentação para o sistema de reconhecimento.

No entanto a remoção dos pontos pertencentes ao plano do chão atrasa um pouco o sistema demorando cerca de  $\sim 80\text{ms}$ , que poderá no futuro ser otimizado caso se revele um obstáculo para o rendimento do guarda-redes.

De uma forma geral apresenta-se na figura 3.11 o estado do programa até esta fase do documento.

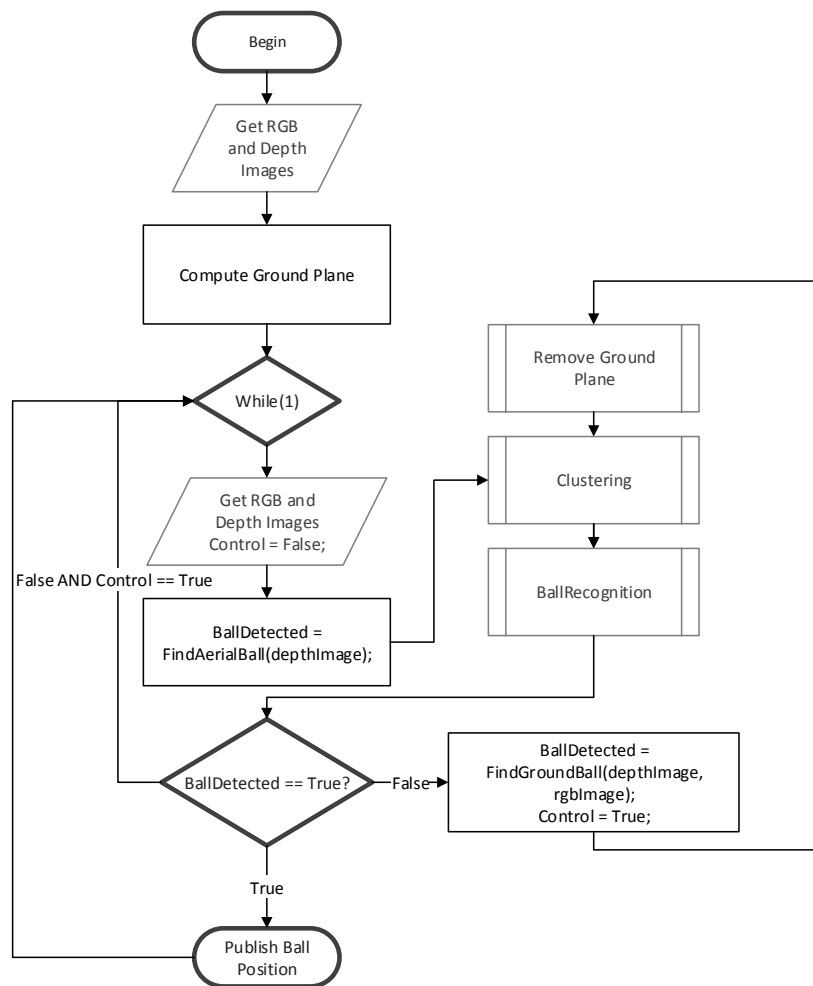


Figura 3.11: Algoritmo do Sistema.





## Capítulo 4

# Algoritmo de Reconhecimento

Neste capítulo aborda-se o processo de reconhecimento que é realizado após a segmentação realizada usando as técnicas especificadas no capítulo anterior.

Para o desenvolvimento de um algoritmo capaz de identificar a presença da bola de futebol recorreu-se à análise das suas propriedades geometricas.

As características que apresentavam melhores resultados às condições que se está a impor ao sistema foram: a área, perímetro, circularidade e o raio do objeto em análise.

Assim para conseguir criar um modelo o mais dinâmico possível e que identifique o valor esperado de acordo com a distância do objeto para as características consideradas. Foi necessário efetuar uma série de aquisições pois verifica-se que os objetos variam as suas dimensões numa imagem captada por uma câmara digital de acordo com a sua distância à câmara, ou seja, quando mais longe mais pequeno parece o objeto e quanto mais perto maior parece o mesmo objeto.

Apresenta-se na figura 4.2 o gráfico resultante das diferentes aproximações polinomiais às características consideradas da bola de futebol.

Para esse efeito utilizou-se a bola apresentada na figura 4.1 para efetuar todas as medições efetuadas durante os testes.



Figura 4.1: Bola de futebol utilizada nos ensaios realizados.

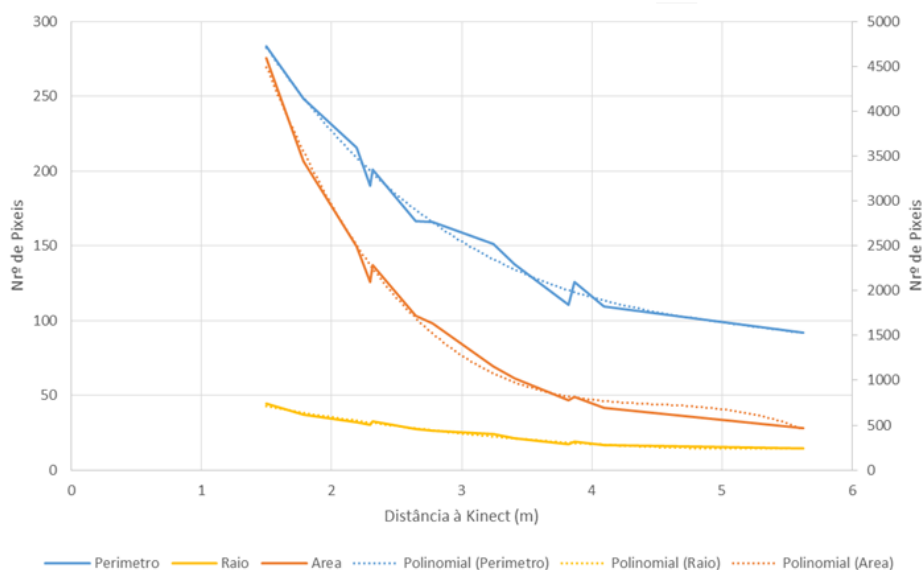


Figura 4.2: Gráfico das variações das diferentes características com a distância à Kinect.

Na tabela 4.1 apresenta-se as funções que modelam cada característica que se obteve do gráfico apresentado anteriormente.

Área	$y = -132,89x^3 + 1788,6x^2 - 8096,5x + 13057$
Perímetro	$y = -2,6906x^3 + 42,524x^2 - 235,36x + 548,86$
Raio	$y = 1,9974x^2 - 21,011x + 69,476$

Tabela 4.1: Funções que modelam as características da bola.

Apresenta-se na figura 4.3 o sistema de reconhecimento que foi desenvolvido para o reconhecimento da bola de futebol.

Observar-se nesta figura que a aplicação trata inicialmente de remover os objetos que ultrapassam os limites máximos e mínimos das características geométricas consideradas para o reconhecimento da bola de futebol que são: área, perímetro, circularidade e raio.

Estes limites foram encontrados experimentalmente tendo em conta as limitações do equipamento usado, ou seja, o sensor de distância é capaz de retornar resultados consistentes dentro de uma gama de distância.

Após esta primeira etapa é obtido dos restantes objetos a sua distância à câmara, utilizando uma máscara 3x3 de forma a evitar ruído que possa surgir num dos pixels do objeto.

Recorrendo às funções que foram calculadas anteriormente e que indicam o valor esperado para as características medidas. Verifica-se cada uma delas a fim de saber se estão dentro de um limite de erro que foi definido experimentalmente.

No final considera-se que o objeto que passe em todas estas etapas é classificado como sendo a bola de futebol que se pretende identificar.

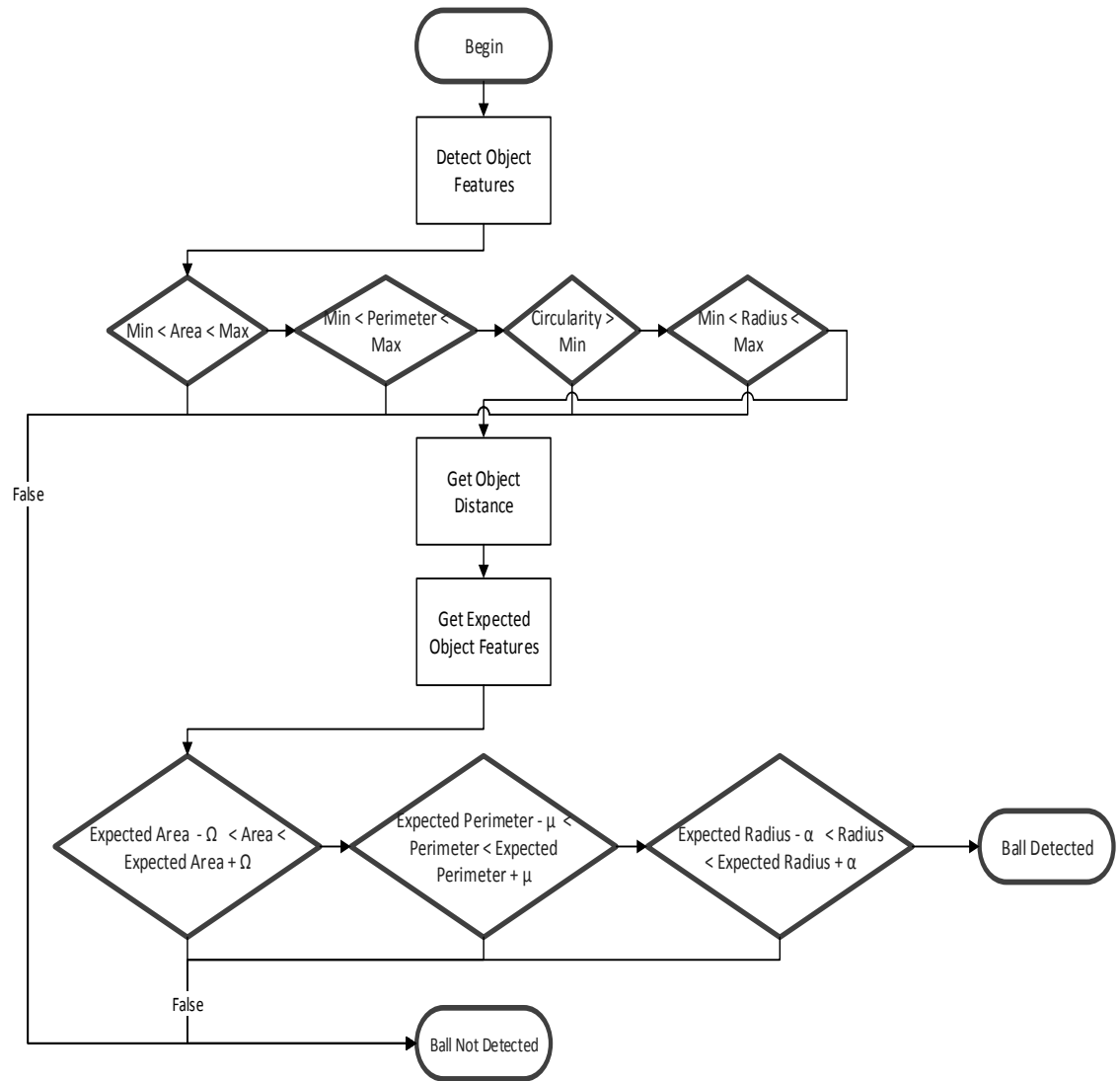
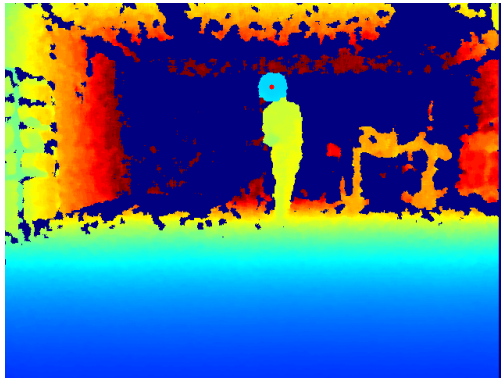


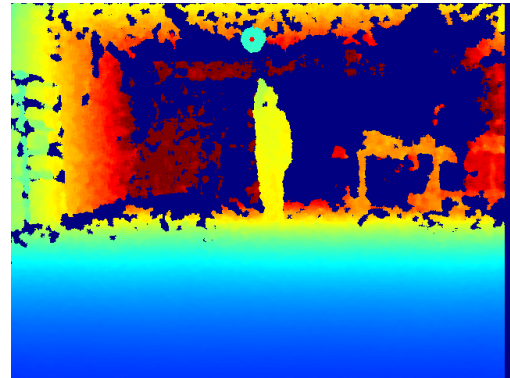
Figura 4.3: Algoritmo de Reconhecimento.

## 4.1 Resultados

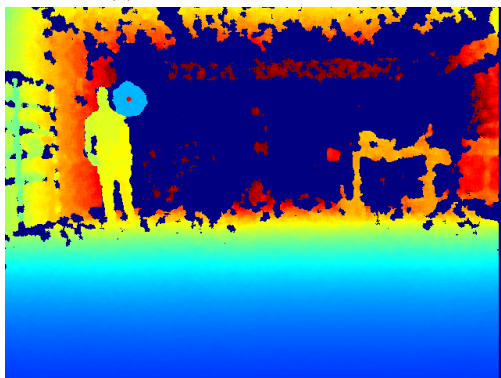
Na figura 4.4 apresenta-se alguns resultados obtidos da detecção e reconhecimento da bola de futebol, onde nas imagens se indica a localização da bola com um ponto vermelho.



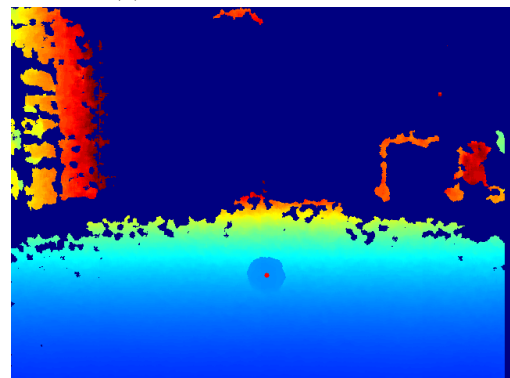
(a) Resultado Bola Aérea 1.



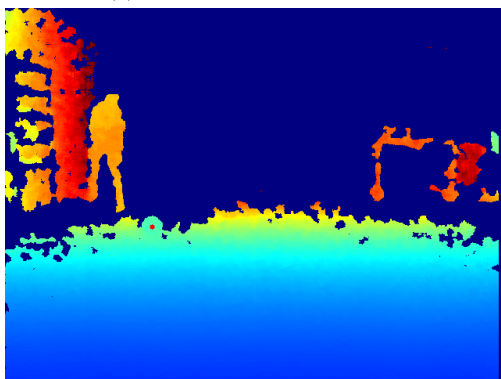
(b) Resultado Bola Aérea 2.



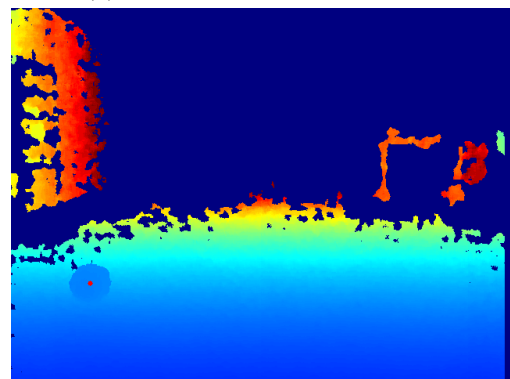
(c) Resultado Bola Aérea 3.



(d) Resultado Bola Não-Aérea 1.



(e) Resultado Bola Não-Aérea 2.



(f) Resultado Bola Não-Aérea 3.

Figura 4.4: Resultado do Algoritmos abordado em 3.1 e 3.2.

## 4.2 Conclusões

Em resumo observa-se que o algoritmo utilizado permitiu o reconhecimento da bola de futebol através da análise das suas características. O tempo total de processamento (incluindo segmentação) fixou-se nos  $\sim 120\text{ms}$ , no pior dos casos (i.e. não existe bola aérea), o que também se revelou ser suficiente para o acompanhamento do movimento da bola embora exista espaço para melhoramento.

Este possível melhoramento recai sobre o aumento da velocidade da segmentação do plano do chão pois sem esta operação o tempo de processamento é reduzido para  $\sim 35\text{ms}$ .

Esta remoção só é efetuada quando não existe uma bola aérea, pois como está especificado no algoritmo [3.11](#) o sistema trata por inicialmente procurar por uma bola aérea e posteriormente, caso esta não exista, procura-se a presença da bola no chão do campo de futebol.



## Capítulo 5

# Filtro de Kalman

Observou-se que o reconhecimento e a correta localização da bola nem sempre é possível em todos os *frames* adquiridos devido ao movimento da bola que dá origem a fenómenos que dificultam a identificação do objeto como sendo a bola de futebol na imagem obtida da cena.

Desta forma foi introduzido no sistema um Filtro de Kalman tridimensional que permite prever onde a bola vai estar no seguinte *frame* tendo em conta os estados anteriores do objeto.

O Filtro de Kalman funciona num processo de duas etapas, a etapa de previsão e a etapa de atualização. Na etapa de previsão 5.1 o filtro baseia a sua saída com os dados obtidos em tempos anteriores recorrendo ao modelo do sistema que se define inicialmente e também à matriz que modela o ruído esperado do processo.

### Previsão

$$\begin{cases} \text{Predicted ("a priori")} \text{ state estimate: } \hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \\ \text{Predicted ("a priori")} \text{ estimate covariance: } \mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \end{cases} \quad (5.1)$$

Onde a variável  $F_k$  é o modelo de transição de estado que é aplicado ao estado anterior do objeto ( $x_{k-1}$ ),  $B_k$  é a matriz que modela o controlo realizado sobre o sistema, a matriz  $Q_k$  é a matriz que modela o erro da medição dos sensores utilizados,  $u_k$  é a matriz de controlo definida pelo utilizador e por fim a variável  $P_k$  é a matriz da covariância do erro de estado.

Na etapa de atualização 5.2 o filtro trata de atualizar as suas matrizes internas tendo em conta a fiabilidade das medidas adquiridas permitindo realizar ajustes de estado pois nem sempre o modelo do sistema definido é totalmente fiel à realidade. Na figura 5.1 apresenta-se uma representação gráfica das etapas do filtro ao longo do tempo.

**Actualização**

$$\begin{cases}
\text{Innovation or measurement residual: } \tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \\
\text{Innovation (or residual) covariance: } \mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\
\text{Optimal Kalman gain: } \mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \\
\text{Updated ("a posteriori") state estimate: } \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\
\text{Updated ("a posteriori") estimate covariance: } \mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}
\end{cases} \quad (5.2)$$

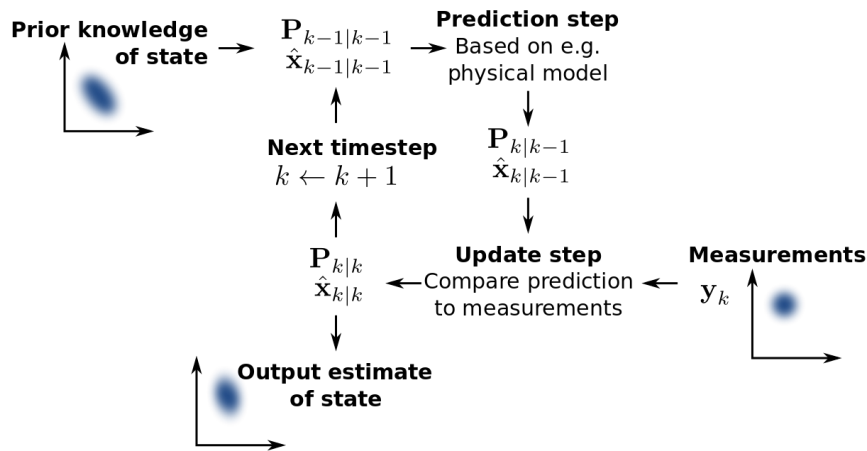


Figura 5.1: O filtro de Kalman guarda os estados estimados do sistema e a incerteza dessa estimativa. A estimativa é atualizada recorrendo ao modelo transitório e às medidas obtidas.  $\hat{\mathbf{x}}_{k|k-1}$  corresponde à estimativa do estado do sistema no tempo  $k$  antes das medidas em  $\mathbf{y}_k$  tenham importância assim o valor de  $\mathbf{P}_{k|k-1}$  corresponde à incerteza da estimativa. [23]

Para o sistema que se definiu neste trabalho foi necessária a criação de dois filtros, ou seja, um para cada caso definido. Nas próximas secções aborda-se os modelos definidos para cada caso.

## 5.1 Modelo para Bola Aérea

Para o caso da bola aérea considerou-se o modelo de aceleração constante pois verifica-se que estando a bola em queda temos sempre presente a aceleração da gravidade.

O estado da bola de futebol ao longo do tempo apresenta-se na equação 5.3 onde temos que  $(x, y, z)$  é a posição no espaço da bola,  $(\dot{x}, \dot{y}, \dot{z})$  é a velocidade da bola nos três eixos e por fim  $(\ddot{x}, \ddot{y}, \ddot{z})$  é a aceleração nos mesmos eixos da bola onde o valor de  $\ddot{z}$  irá ser sempre igual à aceleração da gravidade pois a bola estando em queda este valor é constante.

$$\mathbf{X} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T \quad (5.3)$$



O cálculo do próximo estado da bola pode ser calculado pela equação 5.4 tendo em conta que não há controlo do sistema temos que o valor da matriz  $B$  é igual a zero.

$$X_{k+1} = A \cdot X_k + B \cdot u_k \quad (5.4)$$

Assim a equação 5.4 fica igual à equação 5.5.

$$X_{k+1} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}\Delta t^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot X_k \quad (5.5)$$

O cálculo da posição do objeto pode ser calculado no final pela equação 5.6.

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot X \quad (5.6)$$

## 5.2 Modelo para Bola Não-Aérea

No segundo caso considerou-se o modelo de velocidade constante pois estando a bola em contacto com o chão proveniente de um remate a velocidade irá manter-se-á constante pois a velocidade que a bola pode perder graças ao atrito do chão, distância percorrida, etc. pode ser desprezável neste contexto.

$$X = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \quad (5.7)$$

O cálculo do próximo estado da bola pode ser calculado pela equação 5.8 tendo em conta que não há controlo do sistema temos que o valor da matriz  $B$  é igual a zero.

$$X_{k+1} = A \cdot X_k + B \cdot u_k \quad (5.8)$$

Desta forma a equação 5.8 é equivalente à equação 5.9.

$$X_{k+1} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_k \quad (5.9)$$

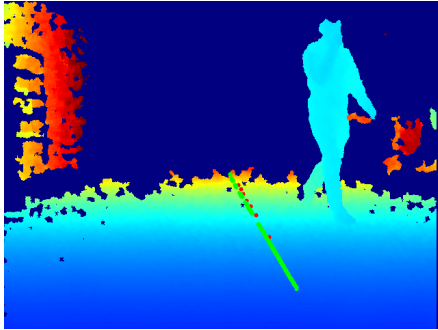
Em suma o cálculo da posição do objeto pode ser calculado no final pela equação 5.10.

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot X \quad (5.10)$$

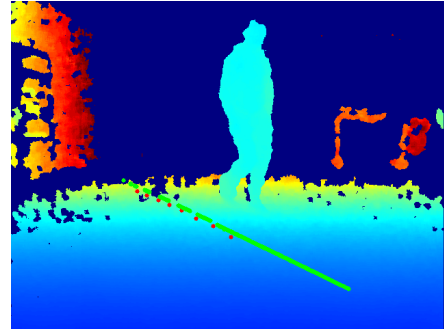
### 5.3 Resultados

Na figura 5.2 e 5.3 apresenta-se o resultado da aplicação do filtro de Kalman (linha de cor verde) e as detecções efetuadas pelos anteriores algoritmos (pontos vermelhos).

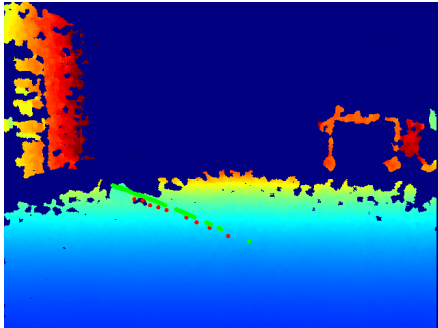
Observa-se que conseguimos efetuar previsões da posição da bola quando não conseguimos efetuar a deteção da bola recorrendo aos algoritmos abordados no capítulo 4.



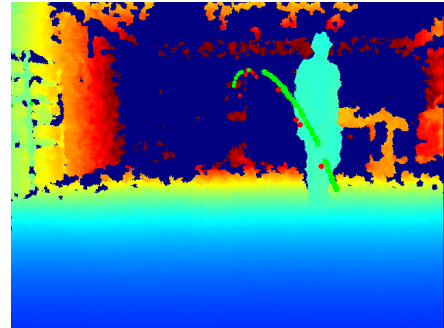
(a) Resultado Kalman Bola Não Aérea 1.



(b) Resultado Kalman Bola Não Aérea 2.

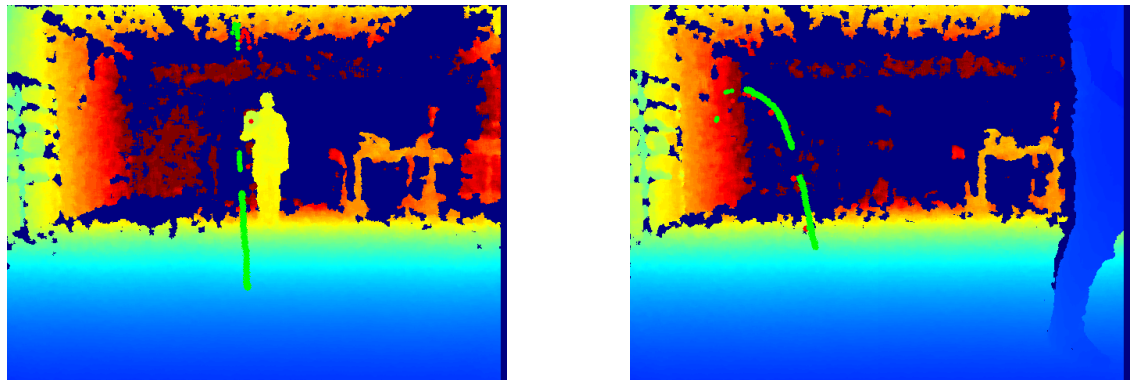


(c) Resultado Kalman Bola Não Aérea 3.



(d) Resultado Kalman Bola Aérea 1.

Figura 5.2: Resultado do Filtro de Kalman na deteção da Bola Aérea e Não Aérea.



(a) Resultado Kalman Bola Aérea 2.

(b) Resultado Kalman Bola Aérea 3.

Figura 5.3: Resultado do Filtro de Kalman na detecção da Bola Aérea.

Na tabela 5.1 apresenta-se o erro médio obtido da diferença da posição real da bola com a posição retornada pelo filtro de Kalman. Pode-se concluir que o modelo definido para o caso da bola não-aérea apresenta melhores resultados do que o caso da bola aérea.

	Número de Pixeis
Kalman Bola Aérea 1	12.3241
Kalman Bola Aérea 2	15.0333
Kalman Bola Aérea 3	13.684
Kalman Bola Não-Aérea 1	3.2083
Kalman Bola Não-Aérea 2	3.6056
Kalman Bola Não-Aérea 3	6.2111

Tabela 5.1: Erro médio da posição real da bola em comparação com a posição retornada pelo filtro de Kalman.

## 5.4 Conclusões

Em síntese pode-se comprovar que o sistema definido tem um resultado final bastante satisfatório permitindo um seguimento do movimento da bola estável permitindo que o guarda-redes intercete a bola antes de esta entrar dentro da baliza.

Observou-se que a definição dos valores das matrizes que dizem respeito ao ruído de processamento e de medição revelaram-se um grande desafio pois têm um elevado impacto no Filtro de Kalman para as suas previsões e correções.



## Capítulo 6

# Conclusões

Apresenta-se neste último capítulo da dissertação as conclusões a que se chegou no final da elaboração deste projeto assim como a apresentação de propostas para o melhoramento deste trabalho.

### 6.1 Conclusões Gerais

Em suma observou-se que a separação do problema da detecção e reconhecimento da bola de futebol em dois casos, bola aérea e bola não aérea, foi uma boa opção pois apresentam problemas diferentes o que permitiu aumentar a velocidade do algoritmo.

O caso da bola aérea é bastante interessante pois verifica-se que enquanto esta está no ar encontra-se desconectada de qualquer elemento da imagem o que permite segmentar rapidamente a imagem por profundidade e efetuar o devido reconhecimento três vezes mais rápido do que para o caso da bola não aérea.

Para o caso seguinte (bola não aérea) verifica-se que esta encontra-se conexa ao chão o que dificulta a segmentação por profundidade pois na criação dos diferentes clusters o algoritmo trata o chão e a bola como sendo o mesmo objeto o que não é verdadeiro. Assim foi necessário efetuar a eliminação dos pontos que correspondem ao plano do chão o que atrasa bastante o algoritmo no entanto não torna esta opção inviável.

Foi inserido o filtro de Kalman pois era necessário preencher as falhas do algoritmo de detecção e reconhecimento pois não é garantido que este consiga detetar a bola em todos os frames adquiridos. Este filtro consegue prever a posição da bola de uma forma eficaz de acordo com o modelo de sistema inserido. O desafio maior que se observou é a modelação do ruído de processo e de medição que pode interferir de forma significativa na sua previsão.

No final deste projeto pode-se constatar que se conseguiu obter bons resultados com o algoritmo desenvolvido e cumprir os objetivos definidos no início da dissertação.

## 6.2 Desenvolvimentos Futuros

Esta dissertação é uma confirmação da utilidade dos sensores RGB-D no futebol robótico, futuramente prevê-se que este algoritmo seja melhorado de modo a detetar e reconhecer a bola de uma forma mais rápida e eficaz.

Para desenvolvimentos futuros propõem-se a melhoria do algoritmo de reconhecimento da bola de futebol que tenha como objetivo de contornar os efeitos provocados na imagem de profundidade pelo movimento rápido da bola.

A utilização de um sensor de profundidade com melhor qualidade também seria uma mais-valia pois desta forma conseguia-se obter resultados mais precisos da bola de futebol evitando algum do ruído que surge na imagem.

# Bibliografia

- [1] Kinect hardware. Accessed on: 10-06-2015. [Online]. Available: <http://www.extremetech.com/wp-content/uploads/2011/11/kinect-for-windows.jpg>
- [2] Kinect specifications. Accessed on: 10-06-2015. [Online]. Available: <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx>
- [3] F. Tombari and L. D. Stefano, “Hough Voting for 3D Object Recognition under Occlusion and Clutter,” *IPSI Transactions on Computer Vision and Applications*, vol. 4, pp. 20–29, 2012.
- [4] D. Lowe, “Object Recognition from Local Scale-Invariant Features,” *IEEE International Conference on Computer Vision*, 1999.
- [5] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006.
- [6] R. B. Rusu, “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments,” pp. 1–4, 2010.
- [7] Narf algorithm. Accessed on: 10-06-2015. [Online]. Available: [http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI\\_tutorial\\_4:\\_3D\\_object\\_recognition\\_\(descriptors\)#NARF](http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI_tutorial_4:_3D_object_recognition_(descriptors)#NARF)
- [8] L. Qiang and L. Feng, “RGB-D sensor based mobile robot SLAM in indoor environment,” *Control and Decision* . . . , pp. 3848–3852, 2014.
- [9] J. a. Batista, “Sistema de Reconhecimento de Objetos para Demonstrador de Condução Robótica Autônoma,” p. 60, 2011.
- [10] M. a. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.
- [11] N. Vaskevicius, K. Pathak, A. Ichim, and A. Birk, “The Jacobs Robotics approach to object recognition and localization in the context of the ICRA’11 Solutions in Perception Challenge,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3475–3481, 2012.

- [12] E. T. H. Zurich, “APPLIED REGISTRATION FOR ROBOTICS Methodology and Tools for ICP-like Algorithms,” no. 21159, 2013.
- [13] W. Guo, T. Du, X. Zhu, and T. Hu, “Kinect-Based Real-Time Rgb-D Image Fusion Method,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXIX-B3, no. September, pp. 275–279, 2012.
- [14] Icp hardware. Accessed on: 10-06-2015. [Online]. Available: <https://taylorwang.files.wordpress.com/2012/04/icp.jpg>
- [15] J. Marcel, “Object Detection and Recognition with Microsoft Kinect,” 2012.
- [16] Yuv color scheme. Accessed on: 10-06-2015. [Online]. Available: [http://upload.wikimedia.org/wikipedia/commons/6/6e/YUV-UV\\_Scaled\\_Y0.5\\_70\\_percent.png](http://upload.wikimedia.org/wikipedia/commons/6/6e/YUV-UV_Scaled_Y0.5_70_percent.png)
- [17] Rgb color scheme. Accessed on: 10-06-2015. [Online]. Available: <http://drmoron.org/images/is-black-a-color/color-cube.png>
- [18] G. Q. Shamik Sural and S. Pramanik, “segmentation and histogram generation using the hsv color space for image retrieval,” pp. 589–592, 2002.
- [19] Hsv color scheme. Accessed on: 10-06-2015. [Online]. Available: <http://i.stack.imgur.com/D5J3v.png>
- [20] Z. Q. Z. Qu and J. W. J. Wang, “A color YUV image edge detection method based on histogram equalization transformation,” *Natural Computation (ICNC), 2010 Sixth International Conference on*, vol. 7, no. Icnc, pp. 3546–3549, 2010.
- [21] A. Sousa, “Arquitecturas de Sistemas Robóticos e Localização em Tempo Real Através Visão,” p. 225, 2003.
- [22] R. M. Haralick, S. R. Sternberg, and X. Zhuang, “Image Analysis Using Mathematical Morphology,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 4, pp. 532–550, 1987.
- [23] Kalman filter concept. Accessed on: 10-06-2015. [Online]. Available: [http://en.wikipedia.org/wiki/Kalman\\_filter#/media/File:Basic\\_concept\\_of\\_Kalman\\_filtering.svg](http://en.wikipedia.org/wiki/Kalman_filter#/media/File:Basic_concept_of_Kalman_filtering.svg)